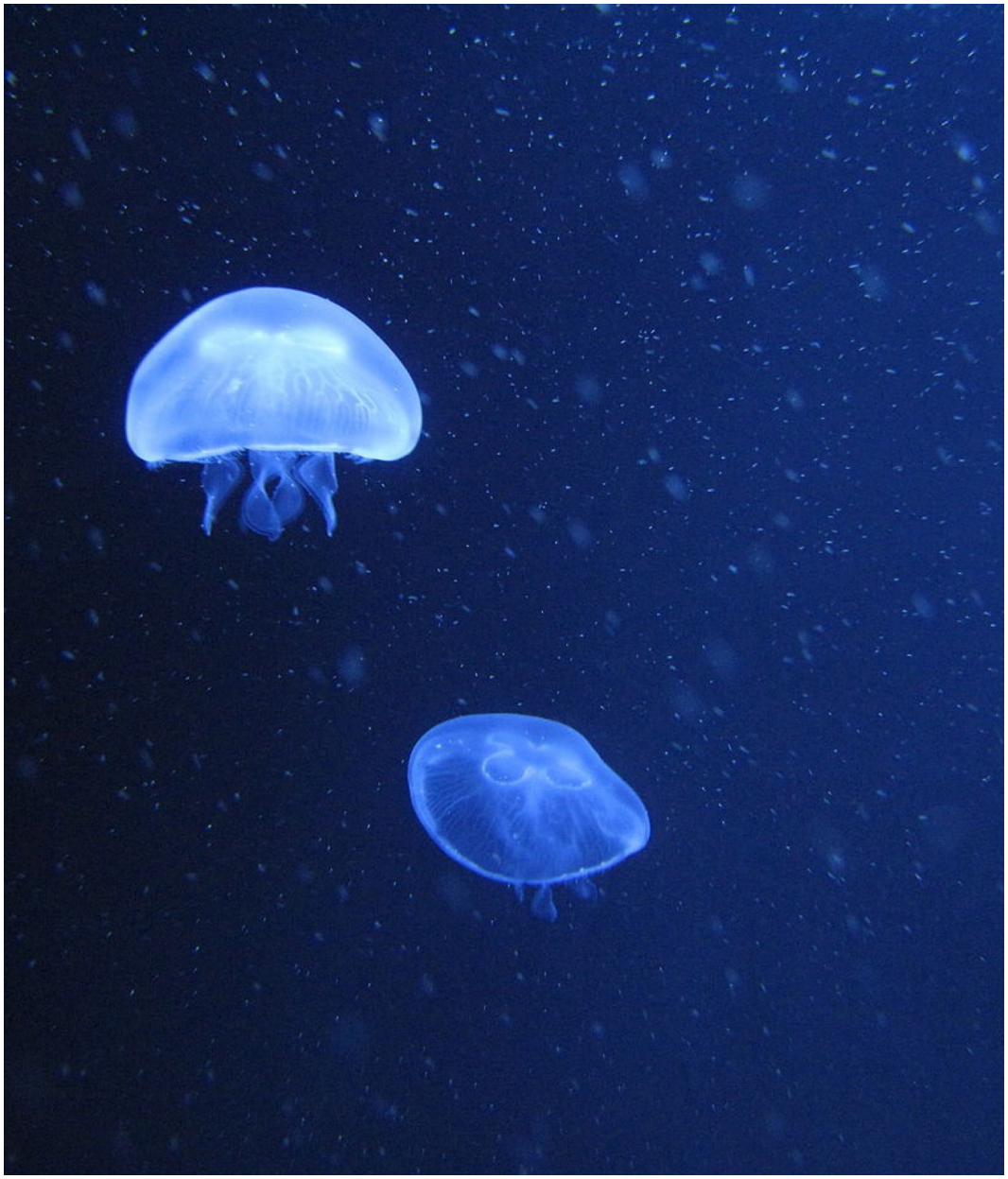


UNIVERSITÉ DE GENÈVE

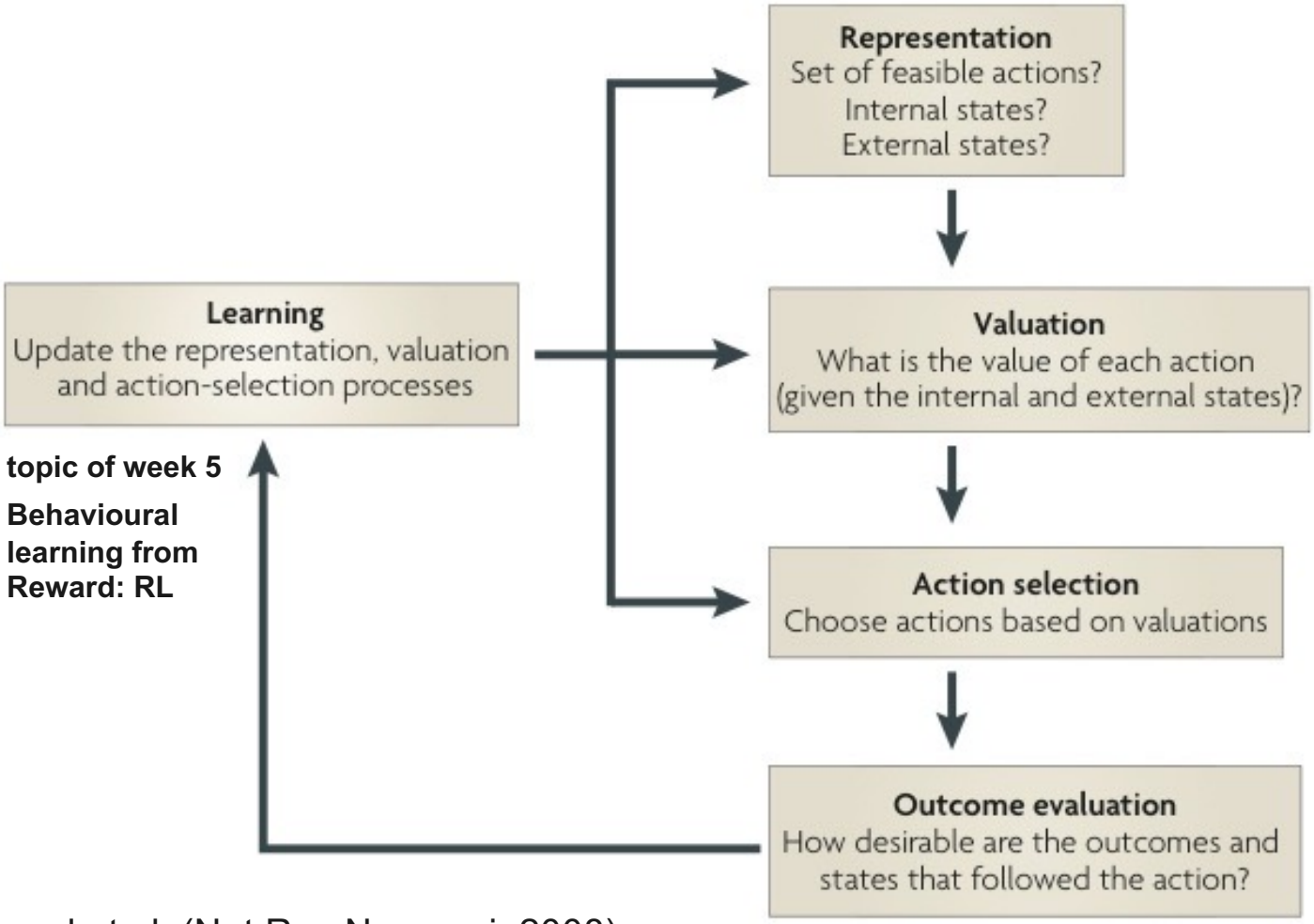
Decision Making Under Uncertainty

Week 5

Reinforcement Learning.
The dopamine reward prediction error hypothesis

Marco Lehmann, PhD

Recap 1/2: Value.



topic of week 5

Behavioural learning from Reward: RL

- **Subjective value:** worth to this person, read from choices or bids
- **Normative value:** what a rational agent ought to choose (expected utility)
- **Incentive value. Motivational valence ("wanting"):** pull towards an outcome; drives approach and effort

- **Experienced value ("liking"). Behavioural reward :** how the outcome feels; pleasure

Rangel et al. (Nat Rev Neurosci, 2008)

Recap 2/2: Value. Today (week 5) Learning

Stage	Concepts (course terms)	What the course has shown	What is open
Representation	Options; states (internal state, e.g. hunger)	Hunger as an internal state (W4 slide 42). We touched "state" or "observations" only implicitly (studies that show a fractal, or a bell ringing, …)	What a "state" is. Learning with or without a model of the world (Tolman; W4 self-study §3).
Valuation	Utility: normative, expected utility (W2). Value: descriptive (subjective), WTP (W2, W3). Incentive value, "wanting" (W2 slide 53). Depends on need/drive and beliefs.	Common currency in vmPFC (Plassmann 2007; Chib 2009; Hare 2010). Changes with state (Gottfried 2003) and beliefs (Plassmann 2008, wine).	Where these values come from.
Action selection	Choice as utility maximisation, "as if" (W2). Pavlovian, habitual, goal-directed systems (W4 slide 47).	Listed only. Chosen vs. unchosen value (Boorman 2009).	How a value becomes an action.
Outcome evaluation	Reward. Experienced value, "liking" (pleasure; W4 slide 33).	Wine liking and mOFC (Plassmann 2008). Dopamine responds to reward relative to expectation (W4 slides 56–58).	How an outcome is compared with the expectation.
Learning	Reward as reinforcement (Thorndike; Skinner). Prediction error.	Preview only (W4 slides 56–61; W4 self-study: Rescorla–Wagner, TD).	All of it: Week 5.

Rangel et al. (Nat Rev Neurosci, 2008)

© Marco Lehmann, 2026

Learning objectives

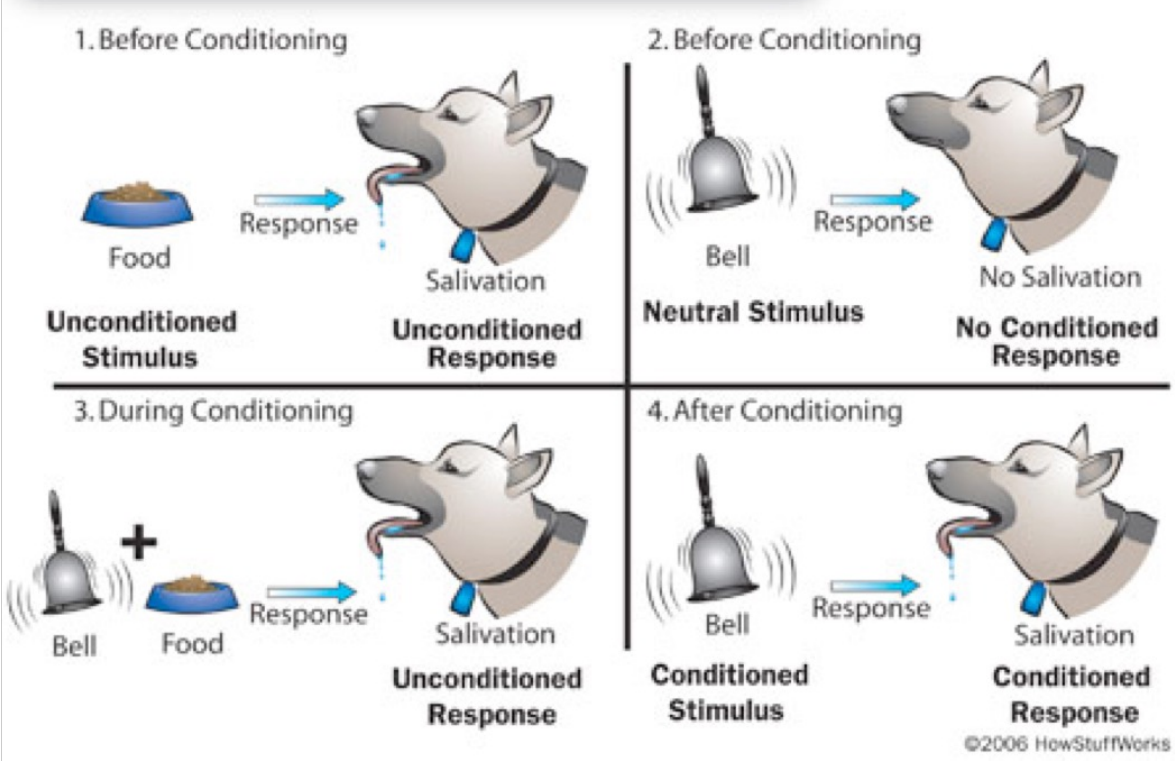
- **Learning from Reward** is a very general mechanism that spans across animal behaviour, neuroscience, and robotics. You understand the basic principles of Reward Learning
- A key signal in learning are **Prediction Errors**. We discuss different forms of prediction errors and study a fundamental **Reinforcement Learning** algorithm
- We review several key experiments since the early 20th century that have led to the current success stories in Reinforcement Learning and the understanding of reward learning in the brain.



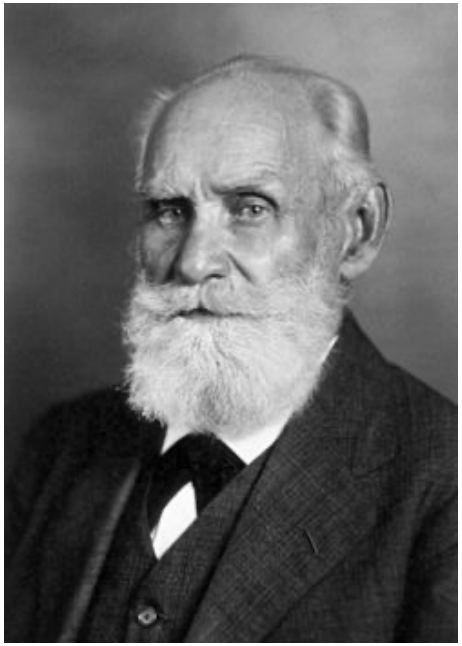
https://youtu.be/9j2a1oAHDL8?si=dwFz82j5WbZEADuY

© Marco Lehmann, 2026

Pavlovian conditioning (classical conditioning)



- Ivan Pavlov, 1849 – 1936,
- Nobel Prize in Physiology or Medicine 1904
- classical conditioning: Association between two stimuli. (**no actions** involved)



© Marco Lehmann, 2026

Rescorla–Wagner is a model of Pavlovian conditioning

- Associate Learning. Note the difference term. How do you interpret it?
- No actions involved !

$$\Delta V_A = \alpha_A \cdot \beta \cdot (\lambda - \Sigma V)$$

Symbol	Name	Meaning
V_A	associative strength of cue A	how strongly A currently predicts the US
ΔV_A	update	the change in V_A on this trial
λ	asymptote	the maximum strength the US can support; $\lambda = 0$ when no US occurs
ΣV	prediction	the sum of V over all cues present on this trial
$\lambda - \Sigma V$		**what is this? see week 4 self-study !**
α_A	salience of cue A	learning rate associated with the cue
β	learning rate of the US	learning rate associated with the US

Next: actions!

- Classical conditioning explains how we learn associations between stimuli, not actions.
- It cannot fully account for voluntary behaviour.
- Operant conditioning extends this by explaining how actions are shaped by their consequences.

© Marco Lehmann, 2026

Behavioural Learning (Operant Conditioning)

A **monkey trained with operant conditioning** assists a paralyzed woman. The monkey was trained to perform tasks such as combing her hair, turning pages in books, and carrying out other helpful actions.



"Psychologie", 16th edition, Zimbardo & Gerrig, page 277

© Marco Lehmann, 2026

Thorndike: «The law of effect»

- https://youtu.be/Vk6H7Ukp6To "Thorndike - Law of Effect"
  («...but **how is a new skill learned**?», «trial and error»)
- Edward Thorndike, 1874 – 1949
- **The law of effect:** "responses that produce a desired effect are more likely to occur again whereas responses that produce an unpleasant effect are less likely to occur again"
  [Thorndike, (1898, 1911) "Animal Intelligence: an Experimental Study of the Associative Processes in Animals"]
- "desired effect" = "Reinforcers" (food, water, sex, sleep, shelter, pleasure, ...)
- "behaviour changes because of its consequences"



Skinner: «Operant Conditioning»

Burrhus Frederic Skinner (1904 –1990), American psychologist

"developed behavior analysis, especially the philosophy of **radical behaviorism**, and founded the experimental analysis of behavior" wikipedia

continued (and unified) the work of Pavlov and Thorndike.



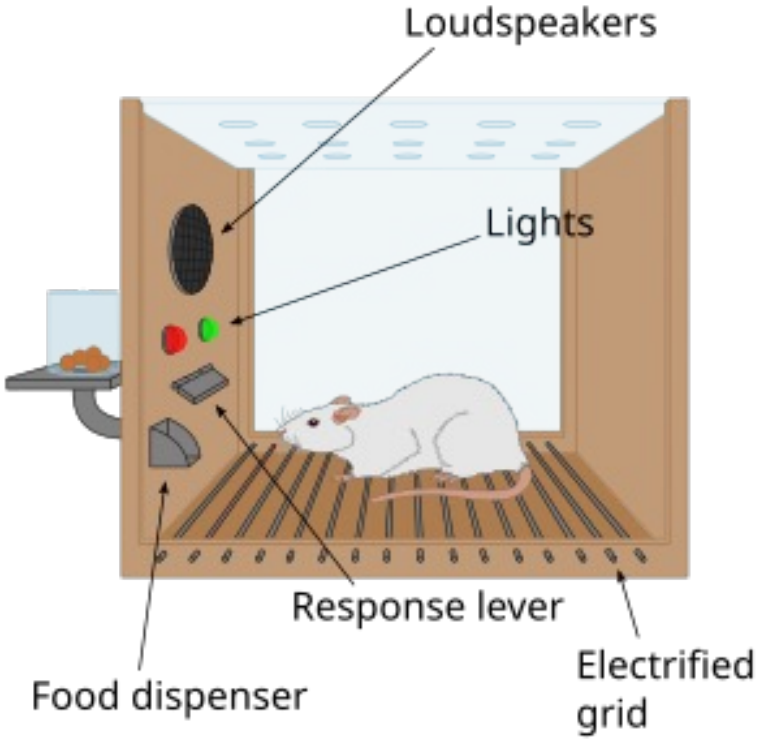
https://en.wikipedia.org/wiki/B._F._Skinner#/media/File:B.F._Skinner_at_Harvard_circa_1950_(cropped).jpg CC BY 3.0

Operant conditioning
"is a type of learning where behavior is shaped by its consequences. **When an action is followed by a reward, we're more likely to repeat it; when it's followed by a punishment, we tend to avoid it.** Psychologist B.F. Skinner showed that this proces, through rewards and consequence, helps explain how habits form, how animals learn tricks, and even how humans are motivated at school and work."
https://www.simplypsychology.org/operant-conditioning.html

© Marco Lehmann, 2026

Skinner Box. Behaviorism

- **Behaviorism**: Psychology as an objective science of observable behaviour (rejected introspection). Dominant mid-20th century paradigm.





An example of Operant Conditioning: a The hungry rats

https://youtu.be/L6MamDfrH-I?si=0lRRShEUOYURmOKC

© Marco Lehmann, 2026

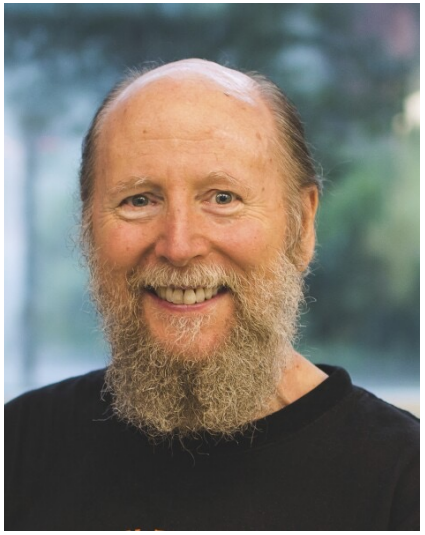
Meanwhile in Engineering and early AI

- 1950s: Richard Bellman studies **optimal control of dynamic systems** (like guiding airplanes, or allocating resources)
  - formulates the Bellman Equation of optimality
  - Develops dynamic programming (DP). DP is a class of algorithms that can answer questions like
    - how to allocate resources now vs. later,
    - how to invest, save, or consume,
    - how to respond to changing prices or risks.
- From the 1950s to 1990s AI researchers developed **Reinforcement Learning (RL)**.
  - Bellman Equation and Dynamic Programming are a fundamental building block.
  - RL formalizes the idea of trial-and-error learning.
  - The field is complex, there are many different algorithms! Today we study one of the main classes: Temporal Difference Learning (**TD-Learning**) with SARSA.

© Marco Lehmann, 2026

Reinforcement Learning

"... in 1979 we came to realize that **perhaps the simplest of the ideas, which had long been taken for granted, had received surprisingly little attention from a computational perspective. This was simply the idea of a learning system that wants something**, that adapts its behavior in order to maximize a special signal from its environment. This was the idea of a “hedonistic” learning system, or, as we would say now, the idea of reinforcement learning." Sutton & Barto, "Reinforcement Learning", 1998



Richard Sutton
Turing Award 2024

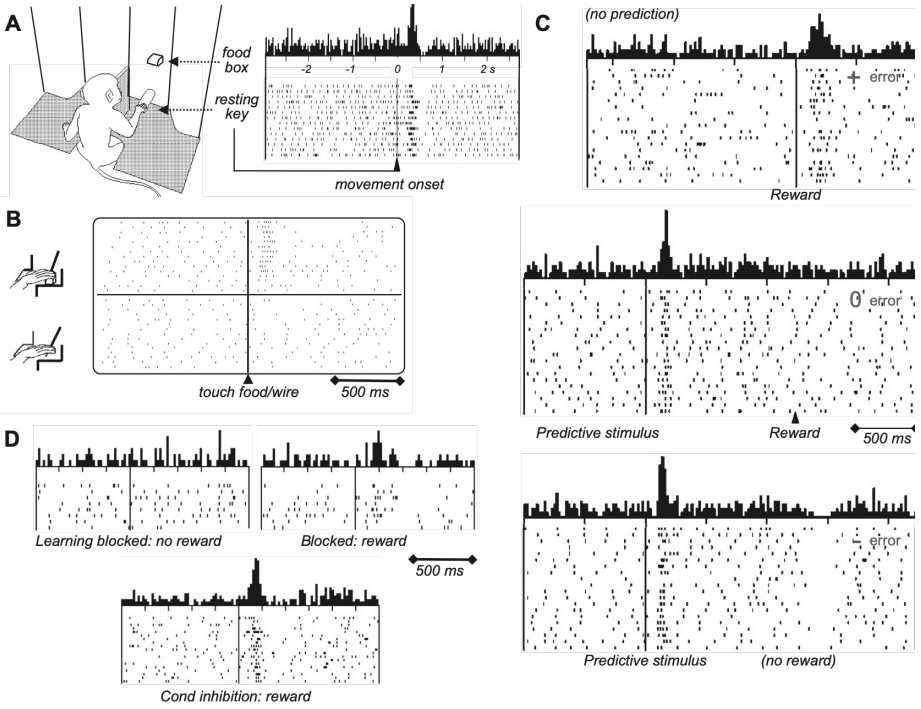


Andrew Barto
Turing Award 2024

© Marco Lehmann, 2026

Meanwhile in neuroscience (1997)

https://brainprize.org/winners/learning-and-reward-2017



see next slide for details

Schultz et al. (1997)



© Marco Lehmann, 2026

© Marco Lehmann, 2026

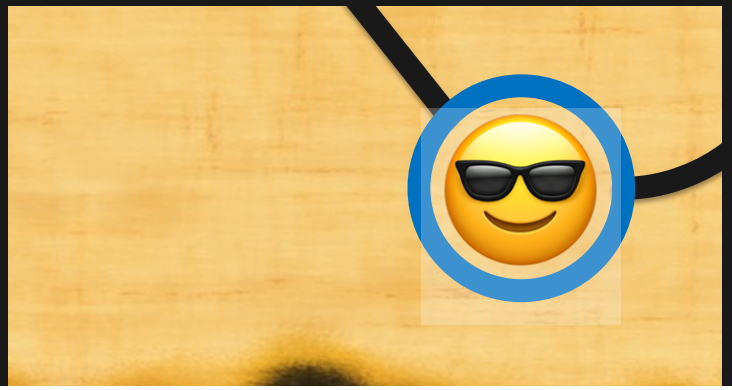
RL Theory. Basic Concepts

- **Reinforcement Learning (RL) formalizes the idea of trial-and-error learning! An agent tries (explores) its options and reinforces the "good" actions.**
- The theory defines a few key concepts:
- **Environment** (an abstract description of the world)
  - States
  - Actions
  - Rewards
  - Transitions

- An RL **agent** tries to "solve" an RL-problem by learning the actions that maximize reward. An agent can be a robot, or a chess-playing software, or an animal in a Skinner box, …
- A **policy** describes the behaviour of an agent. The agent tries to learn the best policy (=the behaviour that maximizes the long term reward).

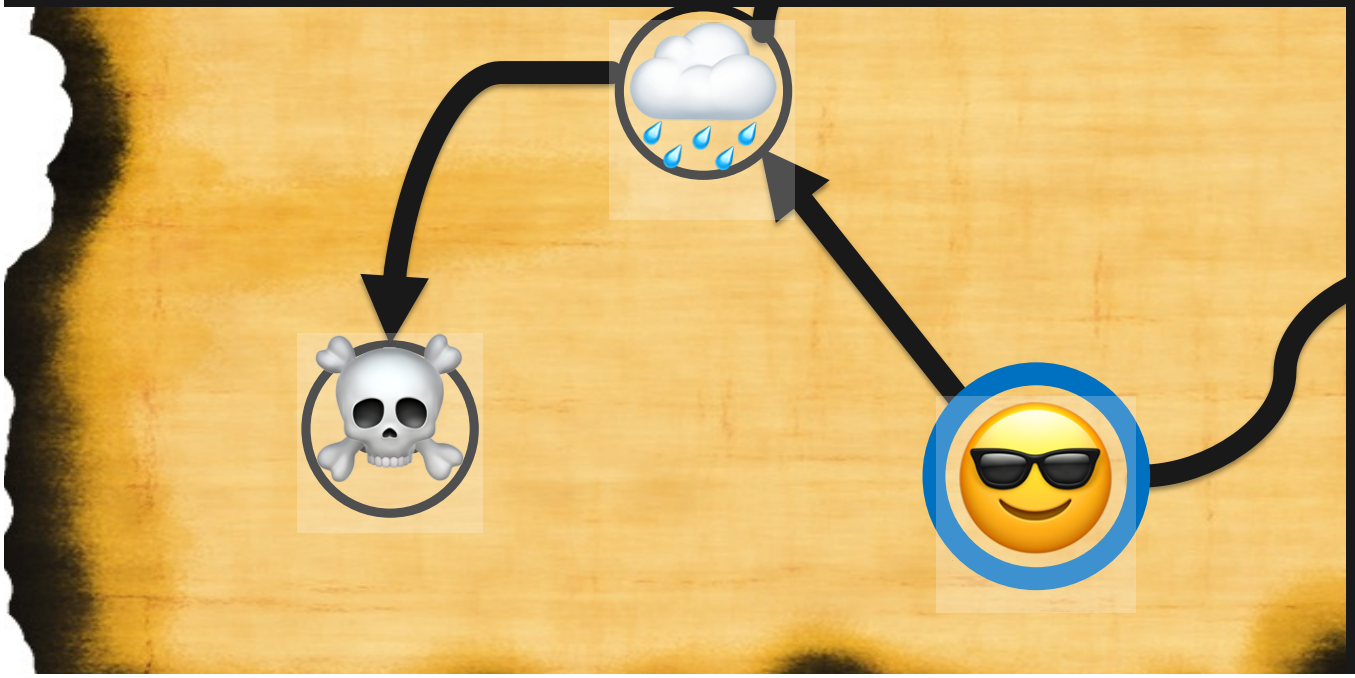
© Marco Lehmann, 2026

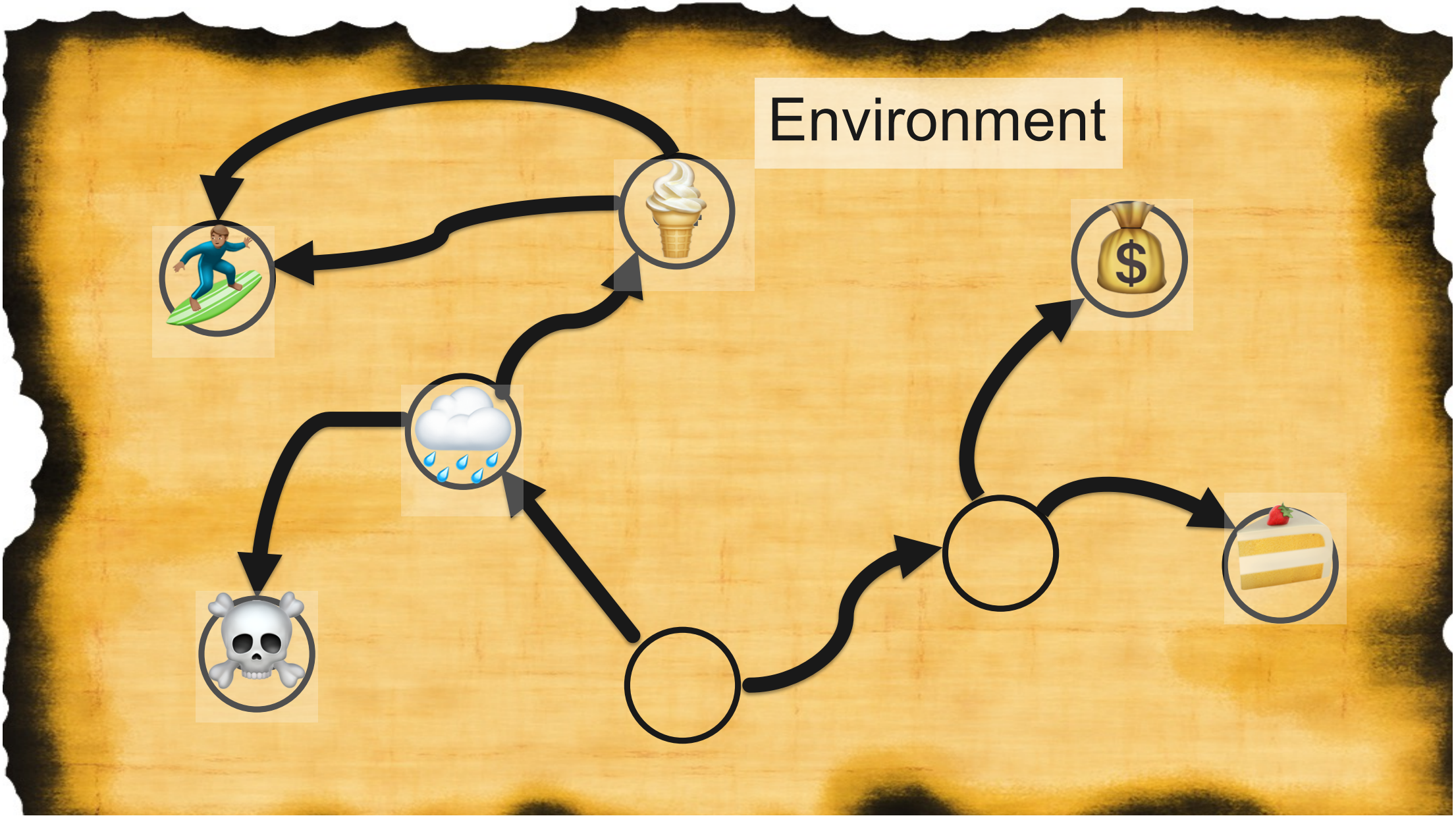
unknown territory. trial-and-error. exploration



unknown territory. trial-and-error. exploration

The exploration – exploitation - dilemma




Environment


Agent
- knows its state
- takes actions
- wants to maximize reward

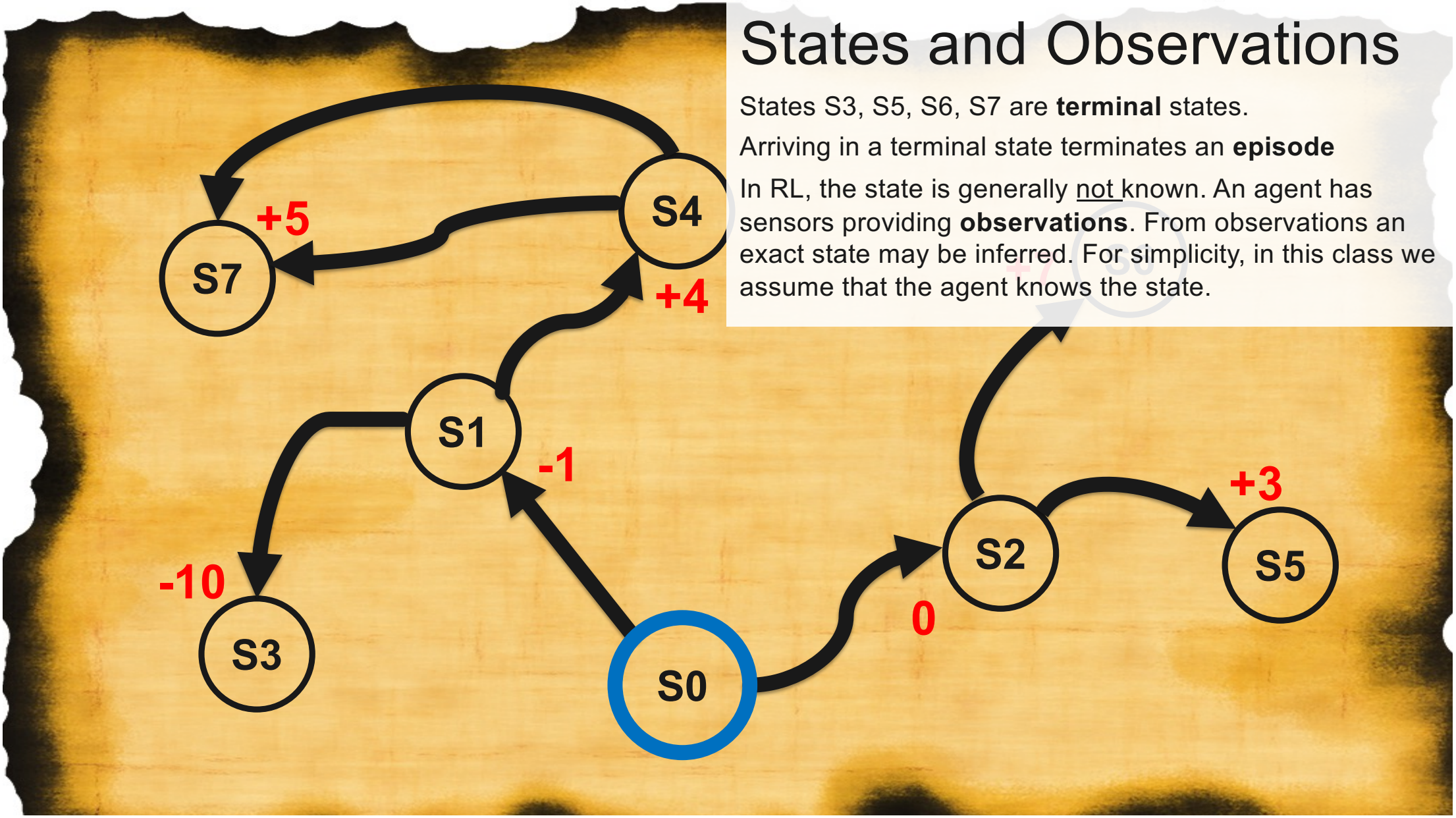

Rewards
+5
+4
+7
-1
+3
-10
0

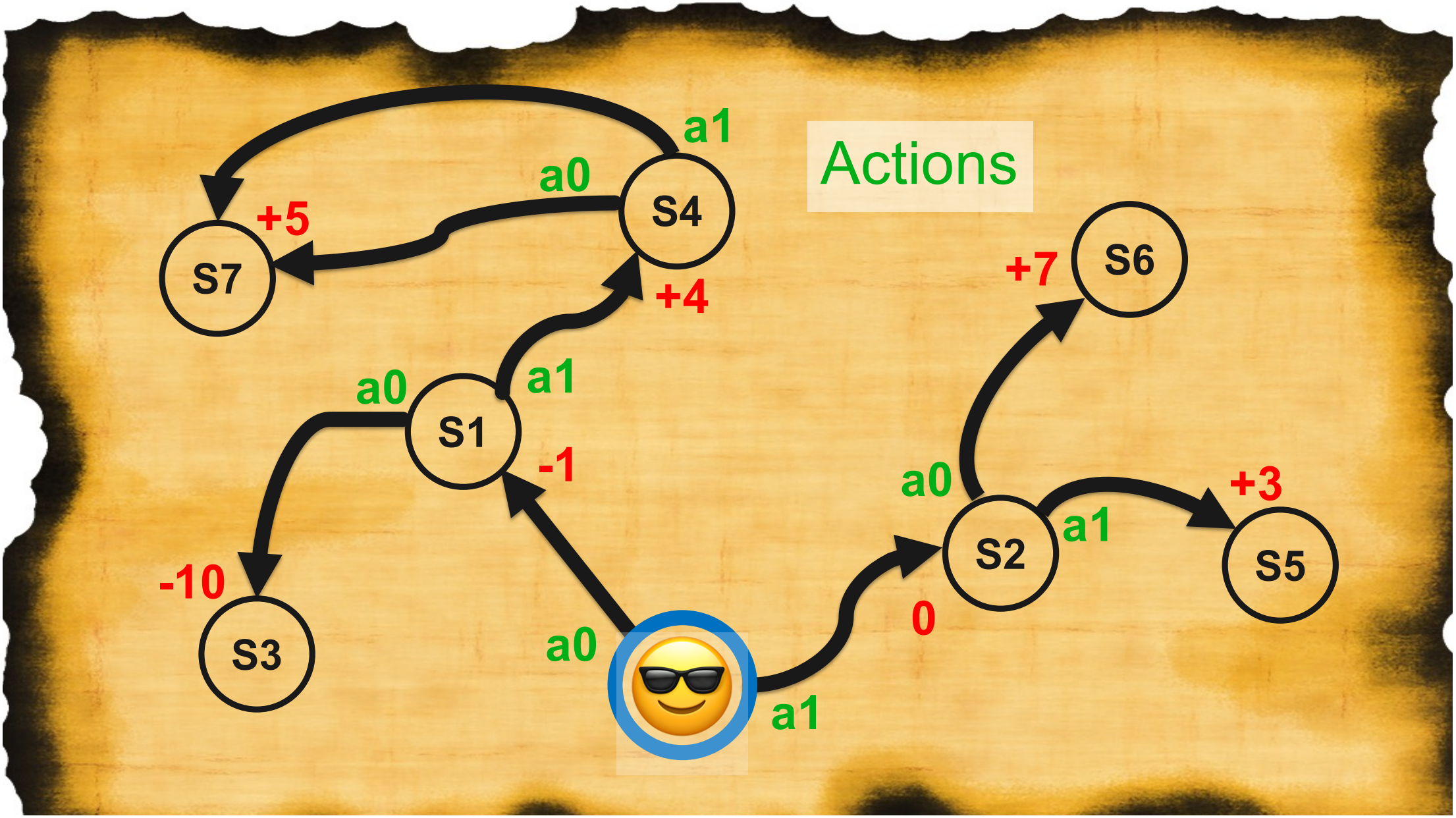
States and Observations

States S3, S5, S6, S7 are **terminal** states.

Arriving in a terminal state terminates an **episode**

In RL, the state is generally not known. An agent has sensors providing **observations**. From observations an exact state may be inferred. For simplicity, in this class we assume that the agent knows the state.




Actions
S1
S2
S3
S4
S5
S6
S7
a0
a1
+5
+4
-1
-10
+7
+3
0

MDP: Markov Decision Process

In the previous example, we've defined a very simple **Markov Decision Process (MDP).** An MDP is composed of 4 entities:

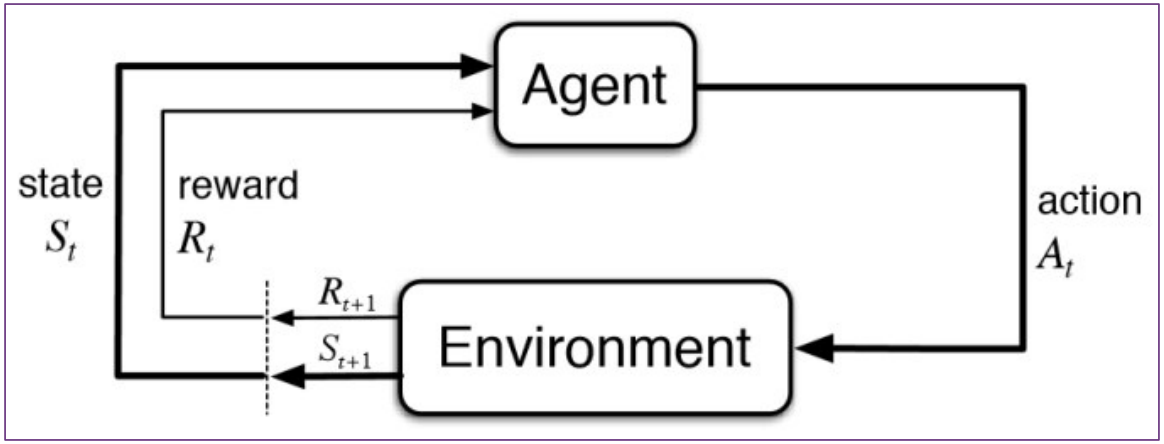
- **S**: the set of **states** { S0, S1, ... , S7 }
- **A**: the set of **actions**: { a0, a1 }
- **R**: the (positive and negative) **rewards** (we will give a precise definition later)
- **P**: The transitions (Here, we used 'Arrows'. More generally, **P** is a set of **transition probabilities**)

- MDPs are a very general mathematical framework. MDPs are used to describe and study a large variety of problems in math, computer-science, psychology, etc. .

© Marco Lehmann, 2026

The general RL framework

- An agent decides which action to take according to its policy π. The action has an effect on the environment. As a result, the environment transitions to the next state and returns a reward. This loop continues either infinitely or until a terminal state is reached.

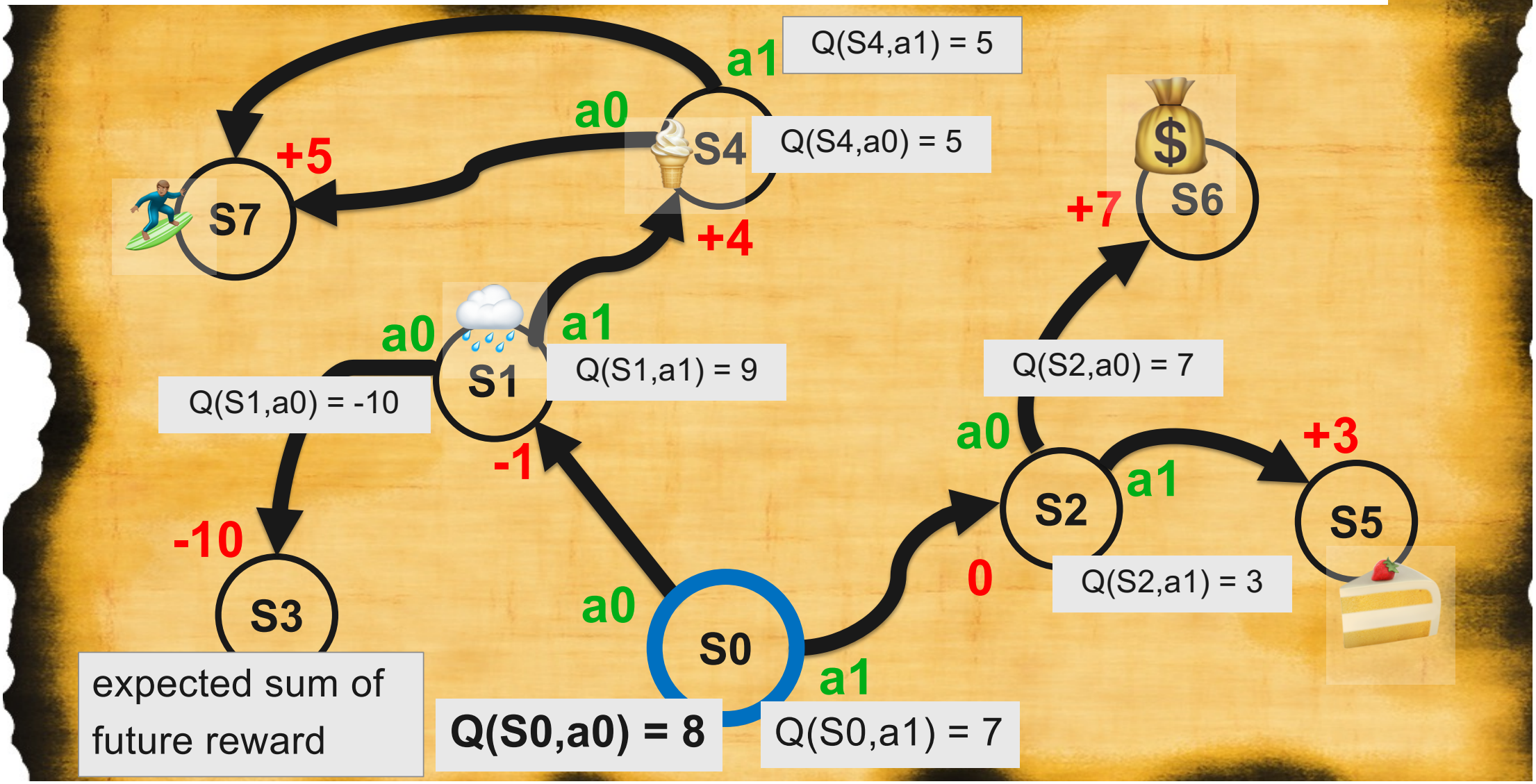


- This is a very abstract (but useful) description! Think about this example: when you (the agent) activate some muscles (=action), then the environment (e.g. the raspberry in your hand) changes its state (its location). A sequence of precisely coordinated muscle activations (a sequence of states and actions) may transition the environment into a rewarding state (e.g. having a raspberry in your mouth, see next slide)

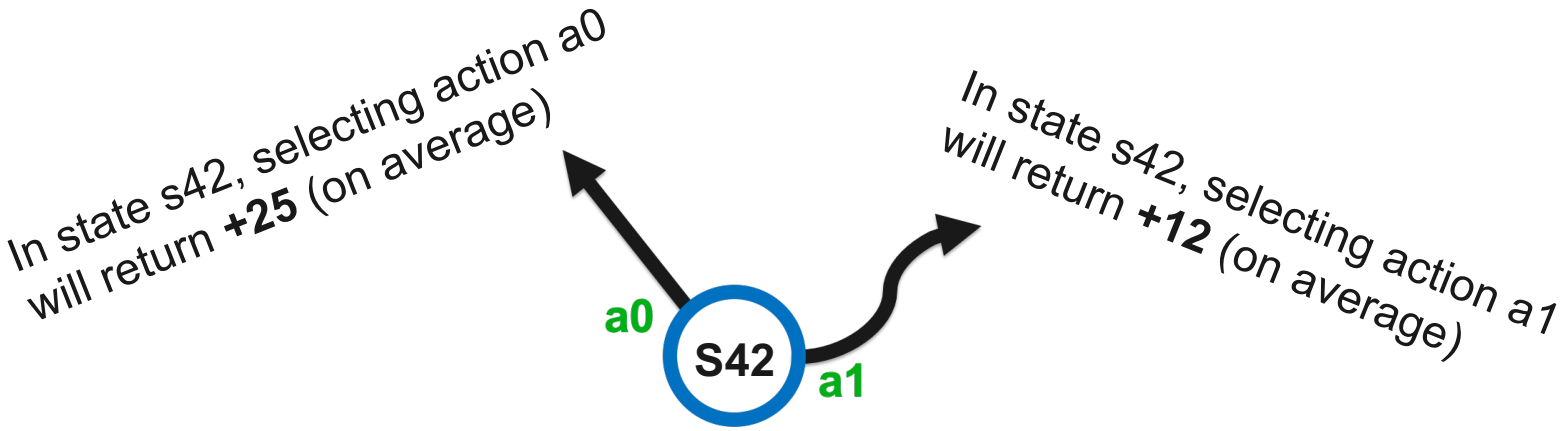
© Marco Lehmann, 2026



RL Solved: For every state/action know its "value"



In RL theory, "Value" is a very abstract concept



- These values are called **Quality-values (Q-values).** They are defined* as an estimate of the expected future rewards. (* a precise definition is given in the reading exercises)
- RL solved: once we know these values for each state-action-pair, taking actions is trivial: at each state s, select the action a with the highest value. Here: Q(S42,a0) > Q(S42,a1) (therefore pick a0)
- (Remember last week? "Strong evidence for a common currency" in the brain)
- The goal of an entire class of RL algorithms is to learn these Q-values.
- How are Q-values learned? Initialize with 0, then explore the environment and update the values. (see next slide)

The SARSA algorithm (State-Action-Reward-State-Action)

Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Initialize $Q(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Choose A from S using policy derived from Q (e.g., ϵ-greedy)
  Repeat (for each step of episode):
    Take action A, observe R, S'
    Choose A' from S' using policy derived from Q (e.g., ϵ-greedy)
    $Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma Q(S',A') - Q(S,A)\right]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

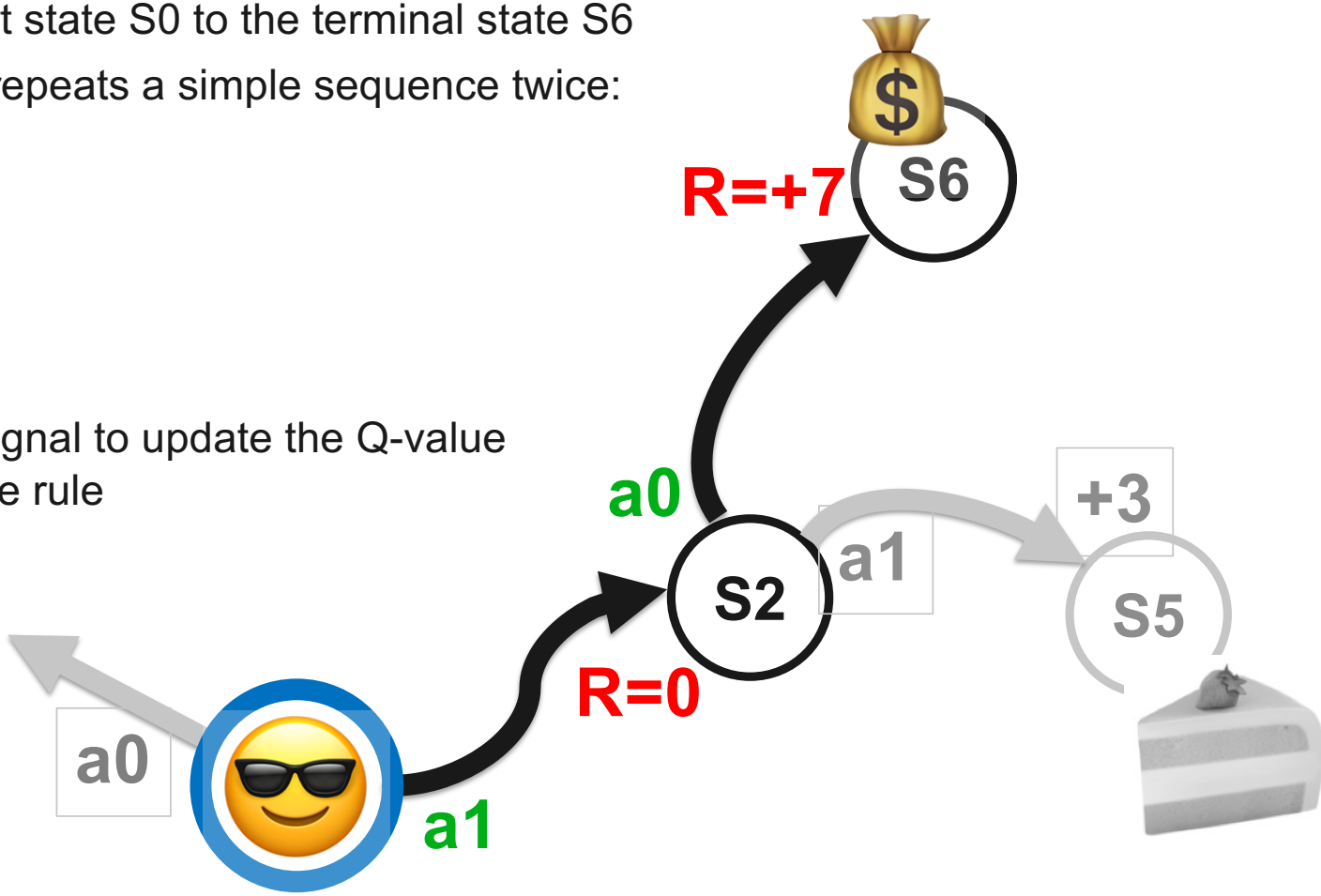
Reward Prediction Error (RPE). aka Temporal Difference (TD) Error

Sutton & Barto 1998

- The RPE is the signal that drives learning
- γ is the "discount factor". We ignore for the moment (set it to 1) and discuss it later in the semester
- α is the learning rate. α = 0 means no learning (keep old predictions). α =1 means old values are overwritten.

Reward Prediction Error (RPE) in a simple 2-step paradigm

- The agent moves from the start state S0 to the terminal state S6
- **Example/Exercise**: An agent repeats a simple sequence twice:
  - Episode 1, step 1: S0, a1
  - Episode 1, step 2: S2, a0
  - Episode 2, step 1: S0, a1
  - Episode 2, step 2: S2, a0
- Reinforcement Learning:
  at each step, use the reward signal to update the Q-value
  according to the SARSA update rule
  Assume $\gamma = 1$ $\alpha = 0.8$



© Marco Lehmann, 2026

Reward Prediction Error (RPE) in a simple 2-step paradigm

Episode 1, step 1:

- current state-action pair: S0, a1
- next state-action pair: s',a' = S2, a0

$$Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma Q(S',A') - Q(S,A)\right]$$

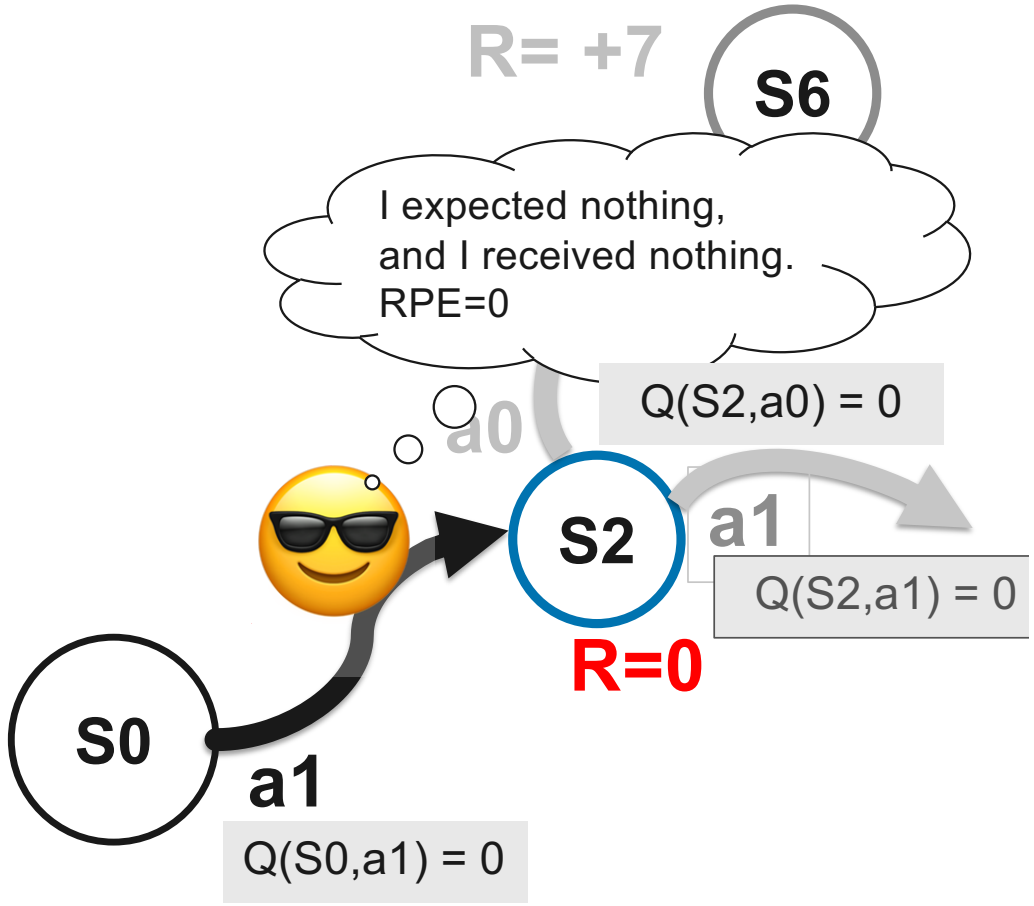
Reward Prediction Error (RPE)

RPE = **0** + Q(S2,a0) – Q(S0, a1)

= 0 + 0 - 0 = 0

Q(S0,a1) ← Q(S0,a1) + α x RPE

Q(S0,a1) = 0 + 0.8 x 0 = 0



© Marco Lehmann, 2026

Reward Prediction Error (RPE) in a simple 2-step paradigm

Episode 1, step 2:

- current state-action pair: S2, a0
- next state-action pair: s',a' = S6, -- (no actions at terminal states)

$$Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma Q(S',A') - Q(S,A)\right]$$

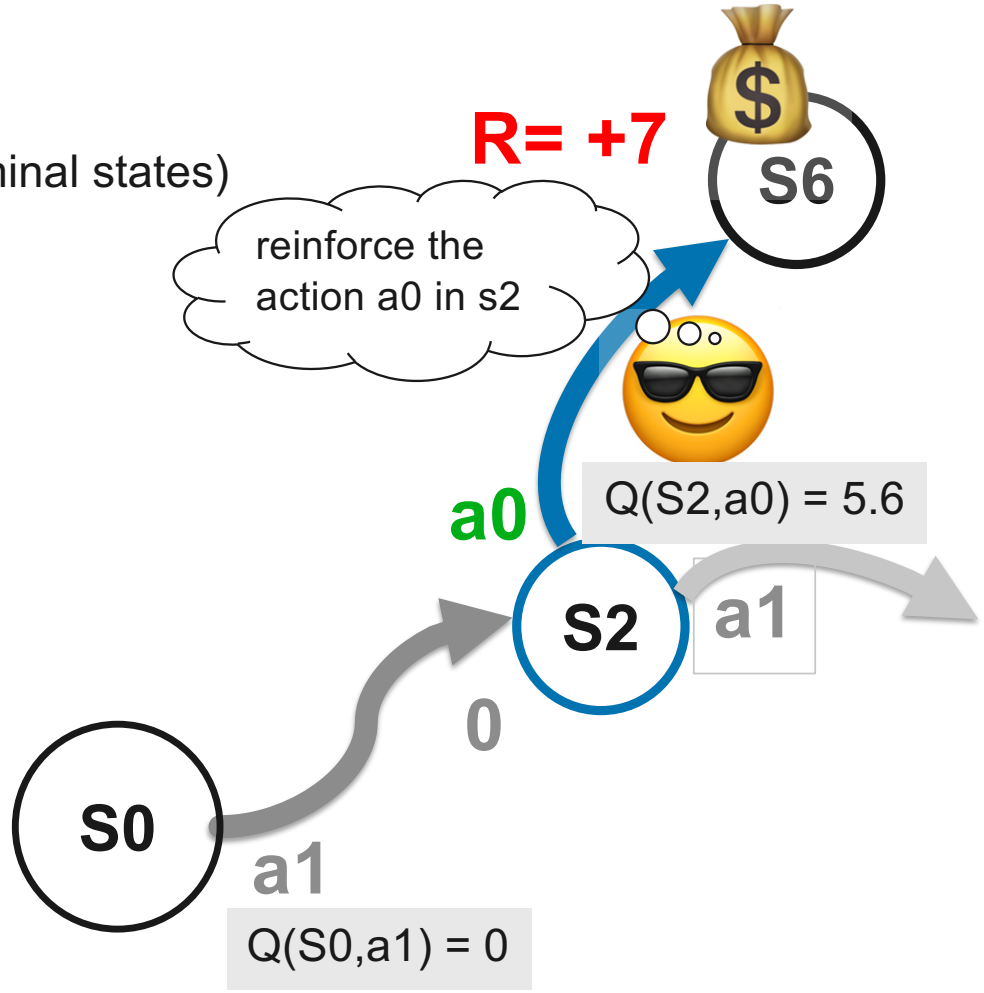
Reward Prediction Error (RPE)

RPE = 7 + Q(S6,--) – Q(S2, a0)

= 7 + 0 - 0 = 7

Q(S2,a0) ← Q(S2,a0) + α x RPE

Q(S2,a0) = 0 + 0.8 x 7 = 5.6



© Marco Lehmann, 2026

Reward Prediction Error (RPE) in a simple 2-step paradigm
Episode 2, step 1: (in-class exercise)

- current state-action pair: S0, a1
- next state-action pair: s',a' = S2, a0

$$Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma Q(S',A') - Q(S,A)\right]$$

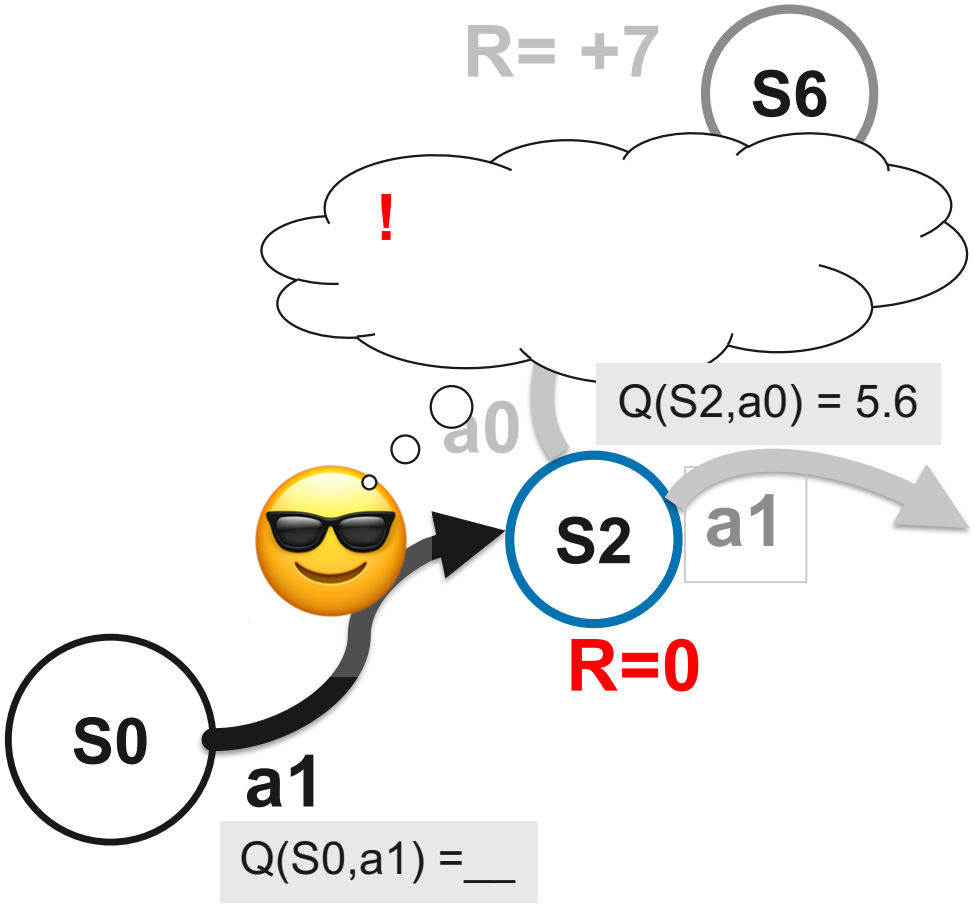
Reward Prediction Error (RPE)

RPE =

=

Q(S0,a1) ← Q(S0,a1) + α x RPE

Q(S0,a1) = _____



© Marco Lehmann, 2026

Reward Prediction Error (RPE) in a simple 2-step paradigm

Episode 2, step 2: (in-class exercise)

- current state-action pair: S2, a0
- next state-action pair: s',a' = S6, -- (no actions at terminal states)

$$Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma Q(S',A') - Q(S,A)\right]$$

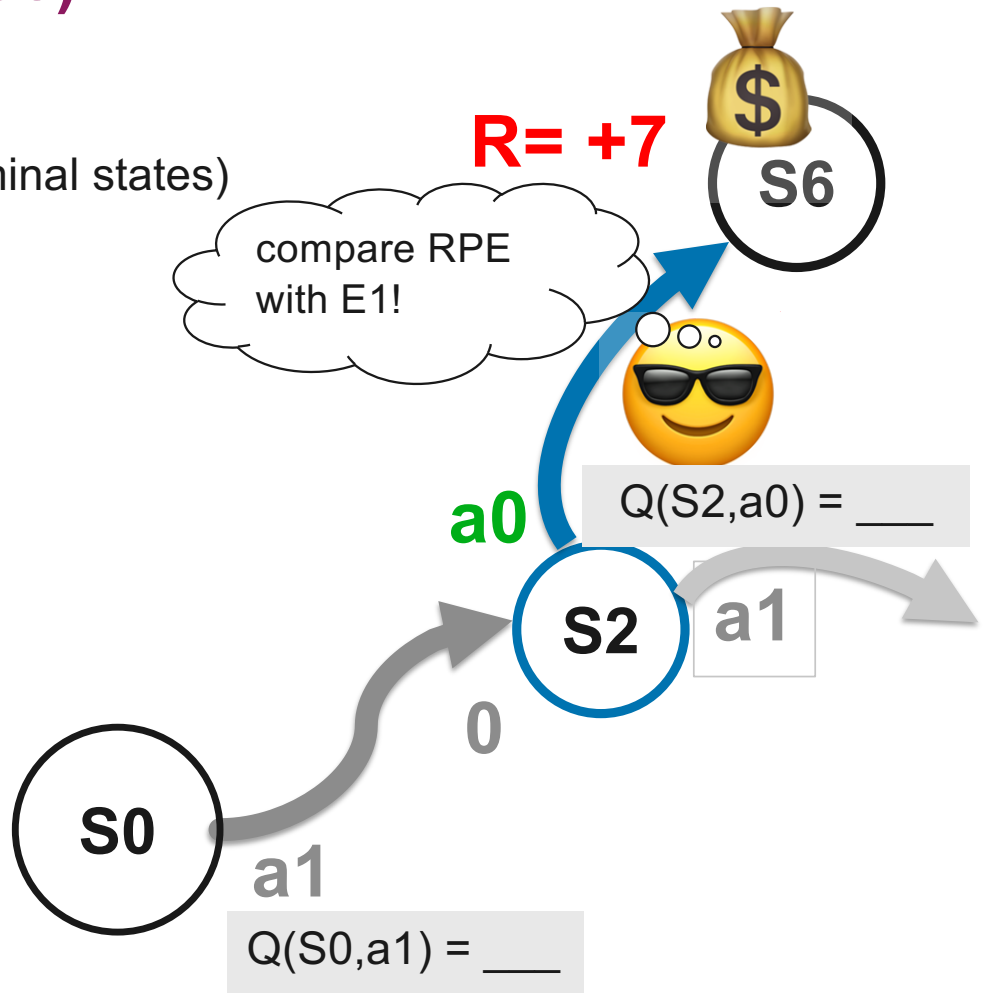
Reward Prediction Error (RPE)

RPE =

=

Q(S2,a0) ← Q(S2,a0) + α x RPE

Q(S2,a0) = _____



© Marco Lehmann, 2026

in-class exercise

- Calculate the steps in Episode 2
- Discuss!
- Then one more example: What if a reward "gets stolen" ? (see next slide)

© Marco Lehmann, 2026

Reward Prediction Error (RPE). Devaluation/Omission

Episode 3, step 2: (in-class exercise)

Someone has stolen the reward at S6. The agent discovers it in step 2:

- current state-action pair: S2, a0
- next state-action pair: s',a' = S6, -- (no actions at terminal states)

$$Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma Q(S',A') - Q(S,A)\big]$$

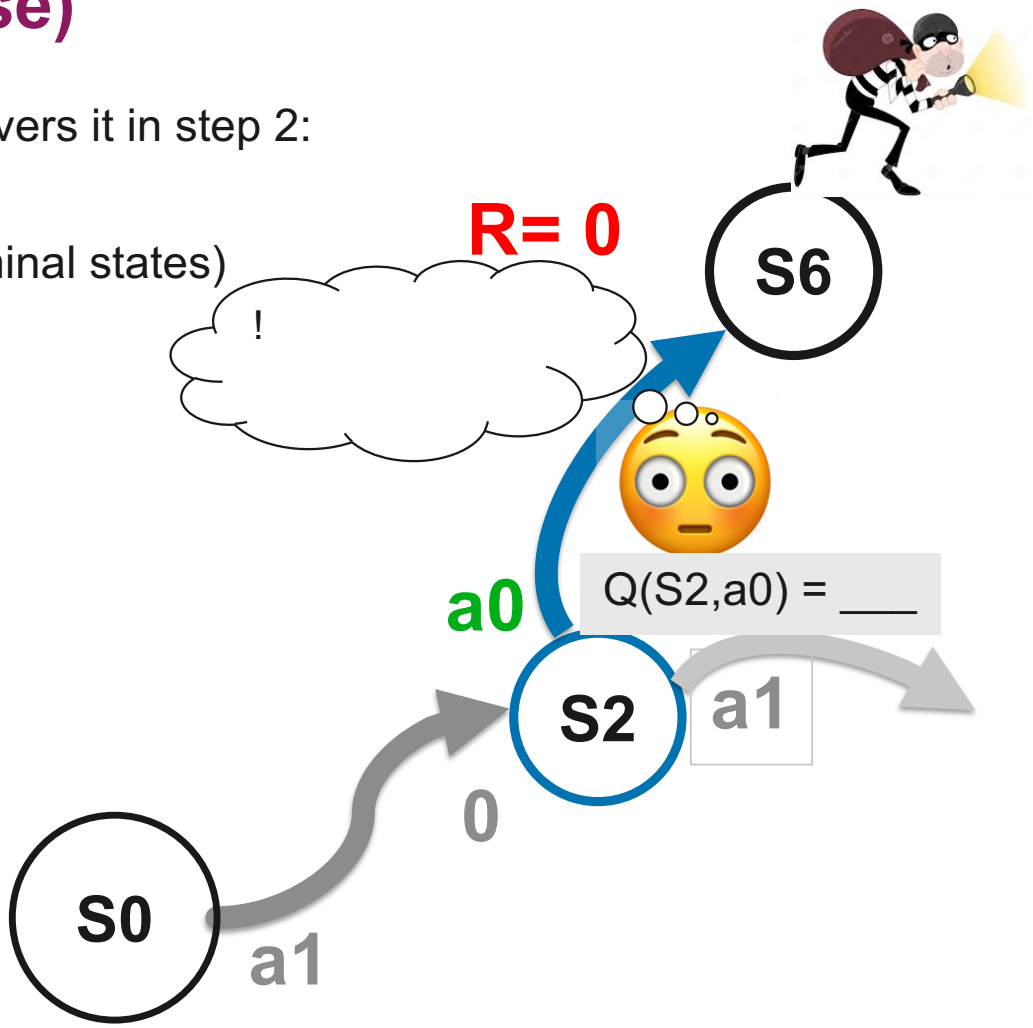
Reward Prediction Error (RPE)

RPE =

=

Q(S2,a0) ← Q(S2,a0) + α x RPE

Q(S2,a0) = _____



© Marco Lehmann, 2026

in-class exercise

- Discuss!

© Marco Lehmann, 2026

Reinforcement Learning. Comments

- The RPE is a trial-by-trial (step-by-step) signal! An agent can learn at every step, not just at the end, when a reward is received or a goal is achieved
- The trial-by-trial signal can be used as a regressor in fMRI studies.
- The RPE approaches 0 as the Q-values correctly predict the (expected) rewards.
- In general (not discussed here), naturally adapts to environments have stochastic transitions and rewards.
- Q-Learning solves the **Credit-Assignment Problem**: "early state/action"-pairs contribute to later reward. Over repetition, the Q-values of these state-action are reinforced.
- If you compare Rescorla-Wagner with RL (SARSA), what are the similarities/differences?

$$\Delta V_A = \alpha_A \cdot \beta \cdot (\lambda - \Sigma V)$$

$$Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma Q(S',A') - Q(S,A)\right]$$

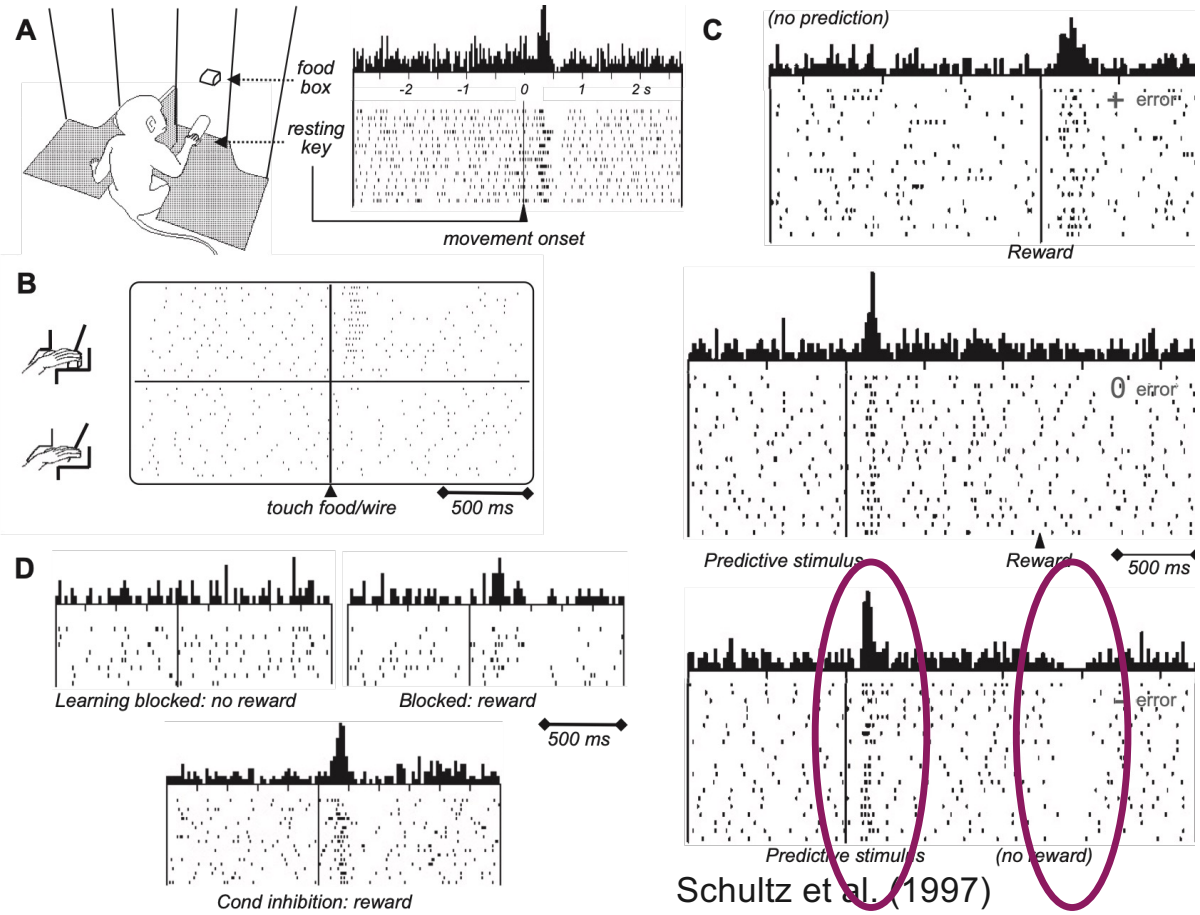
Reward Prediction Error (RPE)

© Marco Lehmann, 2026

RL defines a key learning signal: Reward Prediction Error

Neuroscientists found that signal in the brain:

-> The dopamine reward prediction hypothesis



Schultz et al. (1997)

VTA neurons spiking when an unpredicted reward is delivered

After learning a cue (predictive stimulus) that predicts reward:

- VTA neurons spiking when the cue is shown
- baseline spiking when expected reward is delivered.

After learning:

- VTA neurons spiking when the cue is shown
- below baseline when expected reward is NOT delivered

End of story? No!

- Research has supported the reward prediction hypothesis of dopamine in many experiments
- But Dopamine is more than just RPE
- and learning is more than SARSA

© Marco Lehmann, 2026

nature communications

Explore content | About the journal | Publish with us

nature > nature communications > articles > article

Article | Open access | Published: 13 October 2025

Dopamine dynamics during stimulus-reward learning in mice can be explained by performance rather than learning

Konstantin Bakhurin, Ryan N. Hughes, Qiaochu Jiang, Meghdoot Hossain, Boris Gutkin, Isabella P. Fallon & Henry H. Yin

Nature Communications **16**, Article number: 9081 (2025) | Cite this article

6073 Accesses | **73** Altmetric | Metrics

Abstract

The reward prediction error (RPE) hypothesis posits that phasic dopamine (DA) activity in the ventral tegmental area (VTA) encodes the difference between expected and actual rewards to drive reinforcement learning. However, emerging evidence suggests DA may instead regulate behavioral performance. Here, we used force sensors to measure

Applications: A simplified example from finance

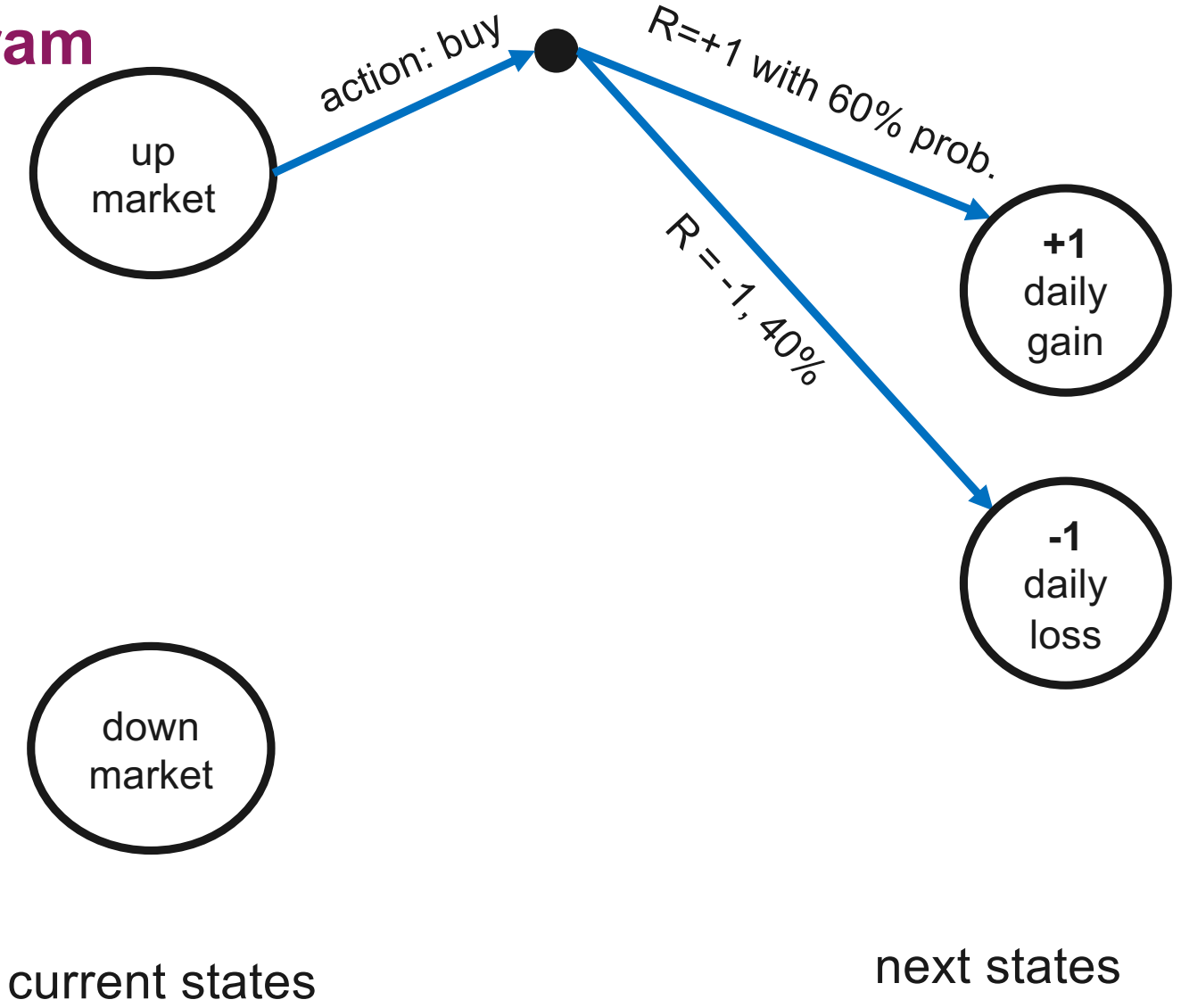
- Scenario: You are an investor trading a single volatile stock over a sequence of trading days. Each day, you must decide whether to **Buy or Sell** one unit of the stock. The market can be in one of two states:
  - **Up (U)** — the price tends to rise,
  - **Down (D)** — the price tends to fall.
- At the end of each day, you observe a discrete return $r \in \{+1, -1\}$ representing gain or loss. Note that the market is volatile: even if it is an up-market, the daily return can be –1 (say, with 40% probability), and on such a day, your reward signal will be –1 if you were buying.
- You must learn about the market and adapt your trading decisions over time.

- The RL model (MDP): state, actions, transitions and rewards. Here is an example:
  - **State**: the market is either in "up" or "down" state. This models the general trend, not certainty
  - **Actions**: the agent can "buy" or "sell"
  - **Transitions are stochastic**! that means: in a general "up" state, the probability that a share goes up is not 100%. Instead: "up" state -> share goes up with 60%, down otherwise. "down" state: -> share goes "down" with 60%, up otherwise
  - **Rewards**: for simplicity, we consider simple, deterministic rewards: +1 for buying a share that goes up, or selling a share that goes down, -1 for the inverse.

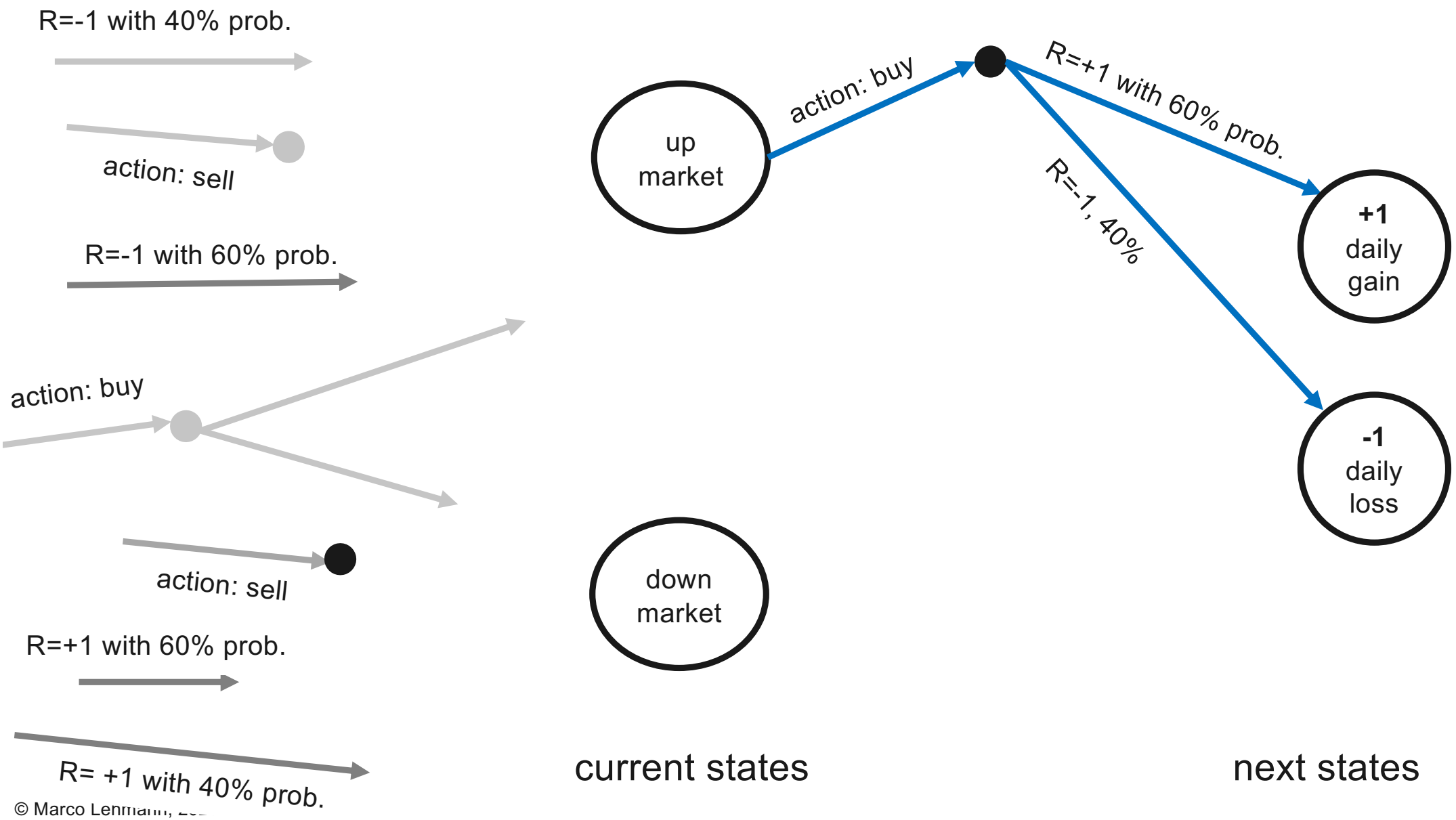
© Marco Lehmann, 2026

Exercise: Backupdiagram

- A backup diagram shows the sequence of states-actions-nextstate-reward. It structures a task and shows how credit is assigned (which state/action led to reward)
- Task: draw the missing elements (states, actions, …)
- see next slide
- find the ppt file on Moodle



© Marco Lehmann, 2026



© Marco Lenmann, 20